Multiplication in Cyclotomic Rings and its Application to Finite Fields

Francisco Argüello

Dept. Electrónica y Computación

Universidad de Santiago de Compostela.

15782 Santiago de Compostela. Spain.

francisco.arguello@usc.es

July 23, 2008

Mailing Address: Francisco Arguello
Dept. Electronica y Computacion
Universidad de Santiago de Compostela
15782 Santiago de Compostela
Spain

PHONE: +34 981 594488 ext. 13556
FAX: +34 981 528012
e-mail: francisco.arguello@usc.es

1

Multiplication in Cyclotomic Rings and its Application to Finite Fields

Francisco Argüello

Abstract

A representation of finite fields that has proved useful when implementing finite field arithmetic in hardware is based on an isomorphism between subrings and fields. In this paper, we present an unified formulation for multiplication in cyclotomic rings and cyclotomic fields in that most arithmetic operations are done on vectors. From this formulation we can generate optimized algorithms for multiplication. For example, one of the proposed algorithms requires approximately half the number of coordinate-level multiplications at the expense of extra coordinate-level additions. Our method is then applied to the finite fields $GF(q^m)$ to further reduce the number of operations. We then present optimized algorithms for multiplication in finite fields with type-I and type-II optimal normal bases.

Keywords: Cyclotomic ring, Finite field, Galois field, Normal basis, Redundant basis, Multiplier.

1   Introduction

Recently, there has been a good deal of interest in developing hardware and software methods for implementing the finite field $GF(q^m)$ arithmetic operations particularly for cryptographic applications [1], [2], [3]. Multiplication in finite fields is a complicated and time-consuming operation that very much depends on how the field elements are represented. A representation of finite fields that has proved useful when implementing finite field arithmetic in hardware is based on an isomorphism between subrings and fields. The main idea is to embed a field in a larger ring, perform multiplication there, and then convert the result back to the field. The ring used is referred to as *cyclotomic*, because has an extremely simple basis whose elements form a cyclic group. Because the dimension of the ring is higher than that of the field, this representation is referred to as redundant. Having in mind the design of efficient arithmetic circuits, it is desirable to find the ring of lowest dimension with the property that the finite field is contained in the ring. This way of representation of finite fields has been explored by various authors [4], [5], [6], [7], [8], [9], [10].

Drolet [5] represents the finite field $GF(2^m)$ as a subring of the *cyclotomic ring* $GF(2)[x]/(x^n + 1)$ with the integer n chosen in such a way that $x^n + 1 \in GF(2)[x]$ contains an irreducible factor of degree m. He shows that this ring representation

1

of elements of the finite field satisfies a generalized Massey-Omura condition and the square of an element can be obtained by applying a specific permutation to the bits of the word representing it. In this line, Geiselmann et al. [6] characterize the smallest n with $GF(2)[x]/(x^n + 1)$ containing an isomorphic copy of $GF(2^m)$.

Some redundant bases can be easily introduced by the normal bases generated with the help of a Gauss period [11], [12]. Gao et al. [7], [8] use Gauss periods for embedding the elements of the finite field in a *cyclotomic field* and, by doing so, they can find the relation/conversion between the redundant basis and the normal basis. This conversion can be done in hardware with almost no cost. There are two types of normal basis generated by Gauss periods with minimal complexity, usually called optimal normal bases (ONBs) of type-I and type-II, respectively. When there exists an ONB, very simple and highly regular multiplier architecture can be obtained using the redundant representation. Recently, Wu et al. [10] have made this idea more explicit and present architectures that are suitable for hardware implementation. The basic idea is to embed the finite field $GF(2^m)$ in the smallest splitting field of $x^n + 1$ over $GF(2)$ and do the arithmetic in this cyclotomic field.

In this paper, we first present an unified formulation for multiplication in cyclotomic rings and cyclotomic fields in that most arithmetic operations are done on vectors. The method is quite generic in the sense that it is not restricted to any special type of ground field. Our algorithms are then applied to the finite fields $GF(q^m)$ with q prime to further reduce the number of operations. The organization of the rest of this paper is as follows: In the next section, we briefly review the cyclotomic rings and fields. In Section 3, we derive a formulation for multiplication in generic cyclotomic rings/fields. We also give the computational complexity of the algorithms in terms of the coordinate-level operations needed. In section 4, we apply the method to the finite fields and then adapt it to two special classes of bases, namely, the type-I and type-II ONBs. Finally, we make a few concluding remarks in Section 5.

2 Cyclotomic Rings and Fields

2.1 Cyclotomic rings

Let F be a field. The set of polynomials with coefficients in the field, $F[x]$, with the usual operations of addition and multiplication of polynomials forms a ring. We can also consider the ring of the polynomials modulo a polynomial $p(x)$. If we let β be the residue class of x, then the elements of $F[x]/p(x)$ can be represented in the form

$$A = a_0 + a_1\beta + a_2\beta^2 + \cdots + a_{n-1}\beta^{n-1}, \quad a_i \in F, \tag{1}$$

where n is the degree of $p(x)$. That is, the elements $1, \beta, \beta^2, \ldots, \beta^{n-1}$ form a true basis for $F[x]/p(x)$.

If the arithmetic is done modulo the polynomial $x^n - 1$ then one obtains the nth *cyclotomic ring* $F[x]/(x^n-1)$. Since a cyclotomic ring satisfies the expression $\beta^n = 1$, the elements $1, \beta, \beta^2, \ldots, \beta^{n-1}$ form a cyclic group of order n with the following multiplication table:

$$\beta \cdot \beta^i = \begin{cases} \beta^{i+1} & \text{if } 0 \le i < n-1 \\ 1 & \text{if } i = n-1. \end{cases} \tag{2}$$

As mentioned in the introduction, the key idea of the representation of $\mathrm{GF}(q^m)$ considered in [5], [6] is to represent the field $\mathrm{GF}(q^m)$ as a subring of $\mathrm{GF}(q)[x]/(x^n-1)$ with $n \ge m$ and do the arithmetic operations over the ring.

Example 1. With the usual addition and multiplication in $\mathrm{GF}(2)[x](x^3+1)$, the residue classes 0, $\beta+1$, β^2+1 and $\beta^2+\beta$ form a subring of $\mathrm{GF}(2)[x]/(x^3+1)$ that is isomorphic to $\mathrm{GF}(2^2)$. The residue class $\beta^2 + \beta$ serves as a multiplicative identity in the subring.

2.2 Cyclotomic fields

On the other hand, the nth *cyclotomic field* [10] over the field F, denoted $F^{(n)}$, is defined to be the splitting field of $x^n - 1$ over F. Let β be a primitive nth root of unity in some extension of F. Then, the elements $1, \beta, \beta^2, \ldots, \beta^{n-1}$ form a cyclic group of order n with the multiplication table (2). $F^{(n)}$ is obtained by adjoining the elements generated by β to F. We may consider the basis $[1, \beta, \beta^2, \ldots, \beta^{n-1}]$ and write elements of $F^{(n)}$ in the form (1).

Since a cyclotomic field satisfies the equation

$$1 + \beta + \beta^2 + \cdots + \beta^{n-1} = 0, \tag{3}$$

the representation is not unique, that is, each n-tuple $(a_0, a_1, \ldots, a_{n-1})$, $a_i \in F$, gives an element of $F^{(n)}$, but different tuples may give the same element. For example, since (3) the two m-tuples $(a_0, a_1, \ldots, a_{n-1})$ and $(a_0+k, a_1+k, \ldots, a_{n-1}+k)$, $k \in F$ both represent the same element.

Example 2. If Q is the field of rational numbers and $n = 3$, then a cyclotomic field $Q^{(3)}$ can be obtained by adjoining a primitive cubic root of unity, β, say $\beta = (-1 + i\sqrt{3})/2$, to the rational numbers Q, and the elements of $Q^{(3)}$ can be written as $A = a_0 + a_1\beta + a_2\beta^2$. Note that equation (3) is satisfied and so such representation is redundant since we can also write $A = b_0 + b_1 i\sqrt{3}$.

As mentioned in the introduction, the basic idea in [10] is to embed the finite field $\mathrm{GF}(q^m)$ in the smallest splitting field of $x^n - 1$ over $\mathrm{GF}(q)$ and do the

arithmetic in this cyclotomic field. Some examples are the redundant bases which can be generated with the help of the Gauss periods [12]. If there exists a normal basis $[\gamma^{q^0}, \gamma^{q^1}, \ldots, \gamma^{q^{m-1}}]$ generated by a Gauss periods of type (m, k), then this normal basis can be expressed in function of the redundant basis $[\beta^0, \beta^1, \ldots, \beta^{n-1}]$ as

$$[\gamma^{q^0}, \gamma^{q^1}, \ldots, \gamma^{q^{m-1}}] = \left[\sum_{i=0}^{k-1} \beta^{q^0 \alpha^i}, \sum_{i=0}^{k-1} \beta^{q^1 \alpha^i}, \ldots, \sum_{i=0}^{k-1} \beta^{q^{m-1} \alpha^i} \right]. \tag{4}$$

where $n = mk + 1$, α is an element of orden k of $\mathbb{Z}_n^\times$ and β satisfies, by construction, equations (2) and (3).

2.3 Multiplication

Let any two elements A, B be represented in the form (1), i.e., $A = \sum_{i=0}^{n-1} a_i \beta^i$ and $B = \sum_{i=0}^{n-1} b_i \beta^i$. Since $\beta^n = 1$ (that is satisfied in both, cyclotomic rings and cyclotomic fields), the product $C = AB$ can be written as,

$$C = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_i b_j \beta^{i+j} = \sum_{j=0}^{n-1} \left(\sum_{i=0}^{n-1} a_i b_{j-i} \right) \beta^j, \tag{5}$$

where the subscript $j-i$ must be read modulo n (i.e., $a_{n+k} \to a_k$ and $a_{-k} \to a_{n-k}$). Then, the coordinates of C can be calculated by

$$c_j = \sum_{i=0}^{n-1} a_i b_{j-i}, \ 0 \le j < n. \tag{6}$$

The resulting algorithm is suitable for a bit-level hardware implementation [10].

3 Algorithm for Multiplication

In this section, we will introduce a vector-level algorithm which essentially eliminates the bit-wide inner products needed by a direct implementation of equation (6). We start from the equation (5) and, using a similar technique to that of [15], write a separate sum with the terms which have equal coordinate indexes,

$$\begin{aligned} C &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_i b_j \beta^{i+j} = \sum_{i=0}^{n-1} a_i b_i \beta^{2i} + \sum_{i=0}^{n-1} \sum_{j=0, j \neq i}^{n-1} a_i b_j \beta^{i+j} \tag{7} \\ &= \sum_{i=0}^{n-1} a_i b_i \beta^{2i} + \sum_{i=0}^{n-1} \sum_{k=1}^{n-1} a_i b_{i+k} \beta^{2i+k}. \tag{8} \end{aligned}$$

4

In the last expression we have used that $\beta^{2i+k} = \beta^{2i+k-n}$ for $2i + k \geq n$ and the subscripts must be read modulo n. Denoting $v = \lfloor (n-1)/2 \rfloor$ and since the multiplication matrix is symmetric, we can write

$$C = \sum_{i=0}^{n-1} a_i b_i \beta^{2i} + \sum_{i=0}^{n-1} \sum_{j=1}^{v} (a_i b_{i+j} + a_{i+j} b_i) \beta^{2i+j} + V, \tag{9}$$

where

$$V = \begin{cases} \displaystyle\sum_{i=0}^{n-1} a_i b_{i+n/2} \beta^{2i+n/2} = \sum_{i=0}^{v} (a_i b_{i+n/2} + a_{i+n/2} b_i) \beta^{2i+n/2} & \text{if } n \text{ even} \\ 0 & \text{if } n \text{ odd.} \end{cases} \tag{10}$$

This equation can be rewritten if we add and subtract the term (for n odd),

$$\sum_{i=0}^{n-1} \sum_{j=0}^{v} a_i b_i \beta^{2i+j} + \sum_{i=0}^{n-1} \sum_{j=0}^{v} a_{i+j} b_{i+j} \beta^{2i+j}. \tag{11}$$

The last sum can be re-indexed, and then one can verify that

$$C = (1 - W) \sum_{i=0}^{n-1} a_i b_i \beta^{2i} + \sum_{i=0}^{n-1} \sum_{j=1}^{v} (a_i + a_{i+j})(b_i + b_{i+j}) \beta^{2i+j} + Z, \tag{12}$$

where

$$W = \begin{cases} \displaystyle\sum_{j=1}^{v} \left(\beta^j + \beta^{-j} \right) + \beta^{n/2} & \text{if } n \text{ even} \\ \displaystyle\sum_{j=1}^{v} \left(\beta^j + \beta^{-j} \right) & \text{if } n \text{ odd,} \end{cases} \tag{13}$$

and

$$Z = \begin{cases} \displaystyle\sum_{i=0}^{v} (a_i + a_{i+n/2})(b_i + b_{i+n/2}) \beta^{2i+n/2} & \text{if } n \text{ even} \\ 0 & \text{if } n \text{ odd.} \end{cases} \tag{14}$$

We can also write $W = \sum_{j=1}^{n-1} \beta^j$. Let $A = \sum_{i=0}^{n-1} a_i \beta^i$ be any ring element. Then one can verify that

$$A(1 - W) = \sum_{i=0}^{n-1} (2a_i - p) \beta^i, \quad \text{with } p = \sum_{j=0}^{n-1} a_j. \tag{15}$$

Applying this expression to (12) and by using a bit of algebra, we can obtain,

$$C = x \sum_{i=0}^{n-1} \beta^i + 2 \sum_{i=0}^{n-1} a_i b_i \beta^{2i} + \sum_{i=0}^{n-1} \sum_{j=1}^{v} (a_i + a_{i+j})(b_i + b_{i+j}) \beta^{2i+j} + Z, \tag{16}$$

5

Multiplier	#Mult	#Doub	#Add	Total
Eqs. (9), (18), rings and fields	n^2	0	$(n-1)n$	$2n^2 - n$
Eqs. (16), (19), rings (general)	$(n+1)n/2$	n	$(3n+1)n/2 - 1$	$2n^2 + 2n - 1$
Eqs. (16), (19), rings (GF(2))	$(n+1)n/2$	0	$(3n-1)n/2 - 1$	$2n^2 - 1$
Eqs. (17), (20), fields (general)	$(n+1)n/2$	n	$3(n-1)n/2$	$2n^2$
Eqs. (17), (20), fields (GF(2))	$(n-1)n/2$	0	$(3n-5)n/2$	$2n^2 - 3n$
Direct [5],[6]	n^2	0	$(n-1)n$	$2n^2 - n$

Table 1: Comparison of cyclotomic ring/field multipliers.

with $x = -\sum_{j=0}^{n-1} a_j b_j$. This equation applies to both cyclotomic fields and cyclotomic rings and to any ground field F.

In the case of cyclotomic fields, we can apply the supplementary relation (3). So, in this case, we obtain the equation

$$C = 2\sum_{i=0}^{n-1} a_i b_i \beta^{2i} + \sum_{i=0}^{n-1} \sum_{j=1}^{v}(a_i + a_{i+j})(b_i + b_{i+j})\beta^{2i+j} + Z. \tag{17}$$

Equations (9), (16) and (17) are the final results. In the next section, we will see how to obtain a vector-level algorithm from these equations.

Table 1 compares the number of coordinate-level operations of the obtained equations with that of a direct implementation of equation (6) (for example, figures 1.a and 1.b [5],[6],[10]). Equation (9) requires the same number of multiplications and additions as the direct implementation. On the other hand, equations (16) and (17) require approximately half the number of coordinate-level multiplications. Although this is achieved at the expense of extra coordinate-level additions, the total number of operations is only slightly higher than that of the direct implementation. Hence, these equations are advantageous for ground fields in which multiplication is more costly than addition. In the particular case where the ground field is GF(2), the number of operations is slightly lower because of $2a_i b_i = 0$. In this case, equation (17) requires the lowest number of operations.

4   Application to finite fields

Finally, we restrict ourselves to the particular case of n odd (for example, for finite fields GF(q) with q prime or power of a prime). In this case and for both cyclotomic rings and cyclotomic fields, equation (9) can be written as

$$C = \sum_{i=0}^{n-1} \left\{ a_i b_i + \sum_{j=1}^{v}(a_{i+j}b_{i-j} + b_{i+j}a_{i-j}) \right\} \beta^{2i}, \tag{18}$$

6

where we have made the change of variables: $i \to i + j$, $j \to -2j$. Moreover, for cyclotomic rings, equation (16) simplifies to

$$C = \sum_{i=0}^{n-1} \left\{ x + 2a_i b_i + \sum_{j=1}^{v} (a_{i+j} + a_{i-j})(b_{i+j} + b_{i-j}) \right\} \beta^{2i}, \qquad (19)$$

with $x = -\sum_{j=0}^{n-1} a_j b_j$ and $v = (n-1)/2$. Lastly, for cyclotomic fields, equation (17) simplifies to

$$C = \sum_{i=0}^{n-1} \left\{ 2a_i b_i + \sum_{j=1}^{v} (a_{i+j} + a_{i-j})(b_{i+j} + b_{i-j}) \right\} \beta^{2i}. \qquad (20)$$

Where, in the particular case of modulo 2 arithmetic, for example GF(2), $2a_i b_i = 0$. From these equations we can develop algorithms in which most arithmetic operations are done on vectors instead of bits. We can make the following considerations:

- The index i runs over all coordinates of the operands, and consequently the operations of multiplication and addition can be done on vectors.

- The subscript $i + k$ represents a cyclic shift of k positions with respect to the reference index i. It is found in the coordinates a_{i+j} and a_{i-j}.

- The square of the basis (β^{2i}) can simply be performed with a permutation since we can write for n odd,

$$
\begin{aligned}
\beta^{2i}|_{i=0}^{n-1} &= [1, \beta^2, \beta^4, \ldots, \beta^{2(n-2)}, \beta^{2(n-1)}] \\
&= [1, \beta^2, \beta^4, \ldots, \beta^{n-1}\beta, \beta^3, \ldots, \beta^{n-2}], \qquad (21)
\end{aligned}
$$

In this last expression, since $\beta^n = 1$, we have applied $\beta^{2j} = \beta^{2j-n}$ if $2j \geq n$. Besides, the inverse permutation represents the realization of a square root operation.

Table 2 shows the data-flow of the coordinates of the variable A during the computation of equation (20). In this table, we can see the cyclic shifts which have to be done in each cycle and the final permutation for obtaining $C = AB$. If this final permutation is not done, we will obtain $D = \sqrt{AB}$. Thus, from equation (18) we have the following algorithm.

Algorithm 1. Multiplication over cyclotomic rings and fields with n odd (equation (18)).
Input: A, B
Output: $D = \sqrt{AB}$, $C = AB$

	a_1	a_2	a_3	a_4	a_5	a_6	a_0
$j = 1$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
	a_6	a_0	a_1	a_2	a_3	a_4	a_5
	a_2	a_3	a_4	a_5	a_6	a_0	a_1
$j = 2$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
	a_5	a_6	a_0	a_1	a_2	a_3	a_4
	a_3	a_4	a_5	a_6	a_0	a_1	a_2
$j = 3$	$+$	$+$	$+$	$+$	$+$	$+$	$+$
	a_4	a_5	a_6	a_0	a_1	a_2	a_3
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
$D = \sqrt{AB} \leftarrow$	d_0	d_1	d_2	d_3	d_4	d_5	d_6
$C = AB \leftarrow$	c_0	c_2	c_4	c_6	c_1	c_3	c_5

Table 2: Data-flow of the coordinates of A during the computation of equation (20) for $C = AB$ and $D = \sqrt{AB}$ with $n = 7$.

1.   Initialize $S_A = A,\ S_B = B,\ v = (n-1)/2$
2. $D = A \odot B$
3.   For $j = 1$ to v {
4. $S_A << 1,\ S_B >> 1$
5. $R = (A \odot S_B) \oplus (B \odot S_A)$
6. $D = D \oplus R$ }
7. $C = \text{sqrt\_perm}(D)$.

From equations (19) and (20) we obtain the following algorithm.

Algorithm 2. Multiplication over cyclotomic rings (equation (19)) and cyclotomic fields (equation (20)) with n odd.
Input: A, B
Output: $D = \sqrt{AB},\ C = AB$

1.   Initialize $S_A = A,\ S_B = B,\ v = (n-1)/2$
2. $D = A \odot B$
//   Lines 3 and 5 apply only to cyclotomic rings
3. $x = -\sum_{i=0}^{n-1} d_i$
4. $D = 2D$
5. $D = D \oplus (x, \ldots, x)$
6.   For $j = 1$ to v {
7. $S_A << 1,\ S_B >> 1$
8. $R = (A \oplus S_A) \odot (B \oplus S_B)$
9. $D = D \oplus R$ }
10. $C = \text{sqrt\_perm}(D)$.

	$a_0 b_0$	$a_1 b_1$	$a_2 b_2$
$j = 1$	$+(a_1 b_2 + b_1 a_2)$	$+(a_2 b_0 + b_2 a_0)$	$+(a_0 b_1 + b_0 a_1)$
$C = AB$	$= c_0$	$= c_2$	$= c_1$

(a)

	$x = a_0 b_0 + a_1 b_1 + a_2 b_2$		
	$x+$	$x+$	$x+$
$j = 1$	$(a_1 + a_2)$	$(a_2 + a_0)$	$(a_0 + a_1)$
	$\times(b_1 + b_2)$	$\times(b_2 + b_0)$	$\times(b_0 + b_1)$
$C = AB$	$= c_0$	$= c_2$	$= c_1$

(b)

Table 3: Multiplication in a cyclotomic ring (example 3). (a) Algorithm 1. (b) Algorithm 2.

In the above algorithms, $\odot$ and $\oplus$ denote coordinate-wise operations, for example, $A \odot B = (a_0 b_0, a_1 b_1, \ldots, a_{n-1} b_{n-1})$, symbols $<<$ and $>>$ denote cyclic shifts, and $C = \text{sqrt\_perm}(D)$ denotes the permutation of coordinates given by $c_{2i \bmod n} = d_i$, $0 \le i < n$.

Next, we apply the above algorithms to multiplication in the finite field $GF(q^m)$. Three cases are considered: the general case (cyclotomic rings), and the two particular cases of finite fields with type-I and type-II ONBs.

A) Cyclotomic rings

This is the more general case (fields are rings with multiplicative inverses) and so we must use Algorithm 1 or Algorithm 2 in full.

Example 3. $GF(2^2)$ is isomorphic to a subring of $GF[x]/(x^3 + 1)$ and so a finite field element can be written in the redundant representation as $A = a_0 + a_1 \beta + a_2 \beta^2$. An isomorphism is given by the embedding $0 \to 0$, $1 \to \beta + \beta^2$, $\alpha \to 1 + \beta$ and $1 + \alpha \to 1 + \beta^2$, where the former is an element of $GF(2^2)$ in polynomial representation and the latter a ring element. Table 3 shows the operations that must be done in a multiplication when using this redundant representation.

B) ONB-I

Some cyclotomic fields can be easily introduced by the normal bases generated by the Gauss periods and, by doing so, one can find the relation/conversion between the redundant basis and the normal basis. In these cases, we can used Algorithms 1 and 2 (the latter without lines 3 and 5).

	a_0b_0	a_1b_1	a_2b_2	a_3b_3	a_4b_4
$j=1$	$+(a_1b_4+b_1a_4)$	$+(a_2b_0+b_2a_0)$	$+(a_3b_1+b_3a_1)$	$+(a_4b_2+b_4a_2)$	$+(a_0b_3+b_0a_3)$
$j=2$	$+(a_2b_3+b_2a_3)$	$+(a_3b_4+b_3a_4)$	$+(a_4b_0+b_4a_0)$	$+(a_0b_1+b_0a_1)$	$+(a_1b_2+b_1a_2)$
$C=AB$	$=c_0$	$=c_2$	$=c_4$	$=c_1$	$=c_3$

(a)

$j=1$	(a_1+a_4) $\times(b_1+b_4)$	(a_2+a_0) $\times(b_2+b_0)$	(a_3+a_1) $\times(b_3+b_1)$	(a_4+a_2) $\times(b_4+b_2)$	(a_0+a_3) $\times(b_0+b_3)$
$j=2$	$+(a_2+a_3)$ $\times(b_2+b_3)$	$+(a_3+a_4)$ $\times(b_3+b_4)$	$+(a_4+a_0)$ $\times(b_4+b_0)$	$+(a_0+a_1)$ $\times(b_0+b_1)$	$+(a_1+a_2)$ $\times(b_1+b_2)$
$C=AB$	$=c_0$	$=c_2$	$=c_4$	$=c_1$	$=c_3$

(b)

Table 4: Multiplication in a type-I ONB (example 4). (a) Algorithm 1. (b) Algorithm 2.

A type-I ONB can be always generated by a Gauss period of type $(m, 1)$. This case is considered in [7], [8], [9], [10], [13], [14], [15], [16], [18]. Here, $n = m + 1$, and a basis for $GF(q^m)$ is $[\beta, \beta^2, \ldots, \beta^m]$ (which is a permutation of the normal basis $[\beta^{q^0}, \beta^{q^1}, \ldots, \beta^{q^{m-1}}]$). The correspondence between finite field elements and cyclotomic field elements is given by

$$
\begin{aligned}
a_1\beta + a_2\beta^2 + \cdots + a_m\beta^m &\longrightarrow 0 \cdot 1 + a_1\beta + a_2\beta^2 + \cdots + a_m\beta^m, \\
(a_1 - a_0)\beta + (a_2 - a_0)\beta^2 + \cdots + (a_m - a_0)\beta^m &\longleftarrow \\
\longleftarrow \quad a_0 \cdot 1 &+ a_1\beta + a_2\beta^2 + \cdots + a_m\beta^m.
\end{aligned} \tag{22}
$$

Example 4. The Gauss period $(4, 1)$ generates an embedding of $GF(2^4)$ in the cyclotomic field $GF^{(5)}$. Table 4 shows the operations that must be done in a multiplication using Algorithms 1 and 2.

Equations (18) and (20) can be simplified in this case since $a_0 = b_0 = 0$. We can also subtract c_0 to $c_1, c_2, ..., c_m$ in accordance with the mapping (22). So equation (18) simplifies to

$$
C' = \sum_{i=1}^{m} \left\{ r + a_i b_i + \sum_{\substack{j = 1, \\ j \neq i, j \neq m+1-i}}^{v} (a_{i+j}b_{i-j} + b_{i+j}a_{i-j}) \right\} \beta^{2i}, \tag{23}
$$

where $r = -\sum_{j=1}^{v} a_j b_{m+1-j} + b_j a_{m+1-j}$ and $C' = \sum_{i=1}^{m}(c_i - c_0)\beta^i$. Also, equation (20) simplifies to

10

$$C' = \sum_{i=1}^{m} \left\{ t + 2a_i b_i + a_{2i} b_{2i} + \sum_{\substack{j = 1, \\ j \neq i, j \neq m+1-i}}^{v} (a_{i+j} + a_{i-j})(b_{i+j} + b_{i-j}) \right\} \beta^{2i},$$

(24)

where $t = -\sum_{j=1}^{v}(a_j + a_{m+1-j})(b_j + b_{m+1-j})$.

C) ONB-II

A Gauss period of type $(m, 2)$ with $n = 2m + 1$ generates a type-II ONB. This case considered in [10], [13], [14], [18], [19], [20]. From equation (4), $\gamma^{q^i} = \beta^{q^i} + \beta^{2m+1-q^i}$, and so, a mapping between finite field elements and cyclotomic field elements can be written as,

$$a_1\beta + a_2\beta^2 + \cdots + a_m\beta^m \longrightarrow$$
$$\longrightarrow \quad 0 \cdot 1 + a_1\beta + a_2\beta^2 + \cdots + a_m\beta^m + a_m\beta^{m+1} + a_{m-1}\beta^{m+2} + \cdots + a_1\beta^{2m},$$
$$(a_1 - a_0)\beta + (a_2 - a_0)\beta^2 + \cdots + (a_m - a_0)\beta^m \longleftarrow$$
$$\longleftarrow \quad a_0 + a_1\beta + a_2\beta^2 + \cdots + a_m\beta^m + a_m\beta^{m+1} + a_{m-1}\beta^{m+2} + \cdots + a_1\beta^{2m}.$$

(25)

Again, $[\beta^1, \beta^2, \ldots, \beta^m]$ is a permutation of the coefficients of the normal basis. In the particular case of $GF(2^m)$, coordinate a_0 in (25) is always zero.

Example 5. The Gauss period $(3, 2)$ generates an isomorphism between $GF(2^3)$ and $GF^{(7)}$. The data-flow of the multiplication is showed in Table 5. In this table, we show the pairs of coordinates over which arithmetic operations are to be performed.

Representation given by (25) has some redundancies which can be eliminated. First, coordinates $c_1, c_2, \ldots, c_m$ are obtained twice, but it can be easy avoided. Applying this simplification to equation (18), we obtain,

$$C'' = \sum_{i=0}^{m} \left\{ a_{s(i)} b_{s(i)} + \sum_{j=1}^{m} (a_{s(i+j)} b_{s(i-j)} + b_{s(i+j)} a_{s(i-j)}) \right\} \beta^{s(2i)},$$

(26)

where the indexes must be read modulo $2m + 1$, and

$$s(i) = \begin{cases} i & \text{if } 0 \leq i \leq m \\ 2m + 1 - i & \text{if } m + 1 \leq i \leq 2m. \end{cases}$$

(27)

We can also subtract c_0 to $c_1, c_2, ..., c_m$ in accordance with the mapping (25). Also, since $a_0 = b_0 = 0$, equation (26) can be simplified to

$j=1$	(1,1)	(2,0)	(3,1)	(3,2)	(2,3)	(1,3)	(0,2)
$j=2$	(2,2)	(3,1)	(3,0)	(2,1)	(1,2)	(0,3)	(1,3)
$j=3$	(3,3)	(3,2)	(2,1)	(1,0)	(0,1)	(1,2)	(2,3)
$C=AB$	c_0	c_2	c_3	c_1	c_1	c_3	c_2

(a)

$j=1$	(2,0)	(3,1)	(3,2)
$j=2$	(3,1)	(3,0)	(2,1)
$j=3$	(3,2)	(2,1)	(1,0)
$C=AB$	c_2	c_3	c_1

(b)

Table 5: Data-flow of multiplication in a type-II ONB (example 5). (a) Original. (b) Simplified.

$$C' = \sum_{i=1}^{m} \left\{ 2y + a_{s(i)}b_{s(i)} + \sum_{j=1,j\neq i}^{m} \left(a_{s(i+j)}b_{s(i-j)} + b_{s(i+j)}a_{s(i-j)}\right) \right\} \beta^{s(2i)}, \qquad (28)$$

where $y = -\sum_{j=1}^{m} a_j b_j$ and $C' = \sum_{i=1}^{m}(c_i - c_0)\beta^i$,

Second, since the data-flow matrix of equation (28) is symmetric, it is only necessary to compute the diagonal and the upper triangular submatrices. In a similar way, equation (20) can be written as

$$C' = \sum_{i=1}^{m} \left\{ 4y + 2a_i b_i + a_{s(2i)}b_{s(2i)} + \sum_{j=1,j\neq i}^{m} \left(a_{s(i+j)} + a_{s(i-j)}\right)\left(b_{s(i+j)} + b_{s(i-j)}\right) \right\} \beta^{s(2i)}. \tag{29}$$

Equations (23), (24), (28) and (29) can be applied to $\mathrm{GF}(q^m)$ with q prime. For comparison, we will consider the particular case of $\mathrm{GF}(2^m)$ and other multipliers as shown in Table 6. This table shows the number of bit operations of these multipliers and the time complexity of multipliers in bit-parallel implementation. The multiplier of [13] is considered to be the first such work published in the open literature, multipliers of [5], [6], [10] use redundant representation, and those of [14], [17], [18], [20] are more recent work and have the best results among the known existing ones.

For multiplication in finite fields using cyclotomic rings, Algorithm 1 requires the same number of arithmetic operations compared to previously published multipliers, while Algorithm 2 requires approximately half the number of AND operations at the expense of extra XOR operations. For multiplication in finite fields with type-I and type-II ONBs, equations (23), (24), (29) and (28) require a total number of operations equal to that of best results known multipliers.

Multiplier	#AND	#XOR	Total	Critical path
Rings, Alg. 1	n^2	$n^2 - n$	$2n^2 - n$	$T_A + \lceil \log_2 n \rceil T_X$
Rings, Alg. 2	$(n^2 + n)/2$	$(3n^2 - n)/2 - 1$	$2n^2 - 1$	$T_A + (1 + \lceil \log_2 n \rceil) T_X$
Rings, redundant [5], [6]	n^2	$n^2 - n$	$2n^2 - n$	$T_A + \lceil \log_2 n \rceil T_X$
ONB-I, Alg. 1	$m^2 + 2m + 1$	$m^2 + m$	$2m^2 + 3m + 1$	$T_A + \lceil \log_2(m+1) \rceil T_X$
ONB-I, Alg. 2	$(m^2 + m)/2$	$(3m^2 + m - 2)/2$	$2m^2 + m - 1$	$T_A + \lceil \log_2(m-1) \rceil T_X$
ONB-I, Eq. (23)	m^2	$m^2 - 1$	$2m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-I, Eq. (24)	$(m^2 + m)/2$	$(3m^2 - m)/2 - 1$	$2m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-I, redundant [10]	$m^2 + m$	$m^2 + m$	$2m^2 + 2m$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-I, [13]	m^2	$2m^2 - 2m$	$3m^2 - 2m$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-I, [17],[18]	m^2	$m^2 - 1$	$2m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-I, [14]	$(m^2 + m)/2$	$(3m^2 - m)/2 - 1$	$2m^2 - 1$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-II, Alg. 1	$2m^2 + m$	$2m^2$	$4m^2 + m$	$T_A + \lceil \log_2(2m+1) \rceil T_X$
ONB-II, Alg. 2, Eq. (26)	m^2	$3m^2 - m$	$4m^2 - m$	$T_A + (1 + \lceil \log_2 m \rceil) T_X$
ONB-II, Eq. (28)	m^2	$(3m^2 - 3m)/2$	$(5m^2 - 3m)/2$	$T_A + \lceil \log_2(2m-1) \rceil T_X$
ONB-II, Eq. (29)	$(m^2 + m)/2$	$2m^2 - 2m$	$(5m^2 - 3m)/2$	$T_A + \lceil \log_2(2m-1) \rceil T_X$
ONB-II, redundant [10]	m^2	$2m^2 - m$	$3m^2 - m$	$T_A + (1 + \lceil \log_2 m \rceil) T_X$
ONB-II, [13]	m^2	$2m^2 - 2m$	$3m^2 - 2m$	$T_A + (1 + \lceil \log_2(m-1) \rceil) T_X$
ONB-II, [18],[20]	m^2	$(3m^2 - 3m)/2$	$(5m^2 - 3m)/2$	$T_A + (1 + \lceil \log_2 m \rceil) T_X$
ONB-II, [14]	$(m^2 + m)/2$	$2m^2 - 2m$	$(5m^2 - 3m)/2$	$T_A + \lceil \log_2(2m-1) \rceil T_X$

Table 6: Comparison of $GF(2^m)$ multipliers.

5 Conclusions

In this paper we have presented an unified formulation for multiplication in cyclotomic rings and fields. From this formulation we can generate optimized algorithms for multiplication. The algorithms are quite generic in the sense that they are not restricted to any special type of ground field. Moreover, in our algorithms, most arithmetic operations are done on vectors. One of the proposed algorithms requires approximately half the number of coordinate-level multiplications compared to the conventional algorithm for multiplication in cyclotomic rings/fields. Although this is achieved at the expense of extra coordinate-level additions, the total number of operations is only slightly higher than that of the conventional algorithm. Hence, the proposed algorithm is advantageous for ground fields in which multiplication is more costly than addition.

Our method has been applied to the finite fields $GF(q^m)$ to further reduce the number of operations. We then presented optimized algorithms for multiplication in finite fields with ONBs of type-I and type-II. In the particular case of $GF(2^m)$ and compared to best results known multipliers in $GF(2^m)$, these proposed ones require the same number of arithmetic operations.

Acknowledgments

References

[1] M.A. Hasan and A.G. Wassal, "VLSI Algorithms, Architectures, and Implementation of a Versatile $GF(2^m)$ Processor", *IEEE Trans. on Computers*, Vol. 49, No. 10, pp. 1065–1073, Oct. 2000.

[2] S. Moon, J. Park, and Y. Lee, "Fast VLSI Arithmetic Algorithms for High–Security Elliptic Curve Cryptographic Applications", *IEEE Trans. on Consumer Electronics*, Vol. 47, No. 3, pp. 700–708, Aug. 2001.

[3] P.H.W. Leong and I.K.H. Leung, "A Microcoded Elliptic Curve Processor Using FPGA Technology", *IEEE Trans. on Very Large Scale Integration Systems*, Vol. 10, No. 5, pp. 550-559, Oct. 2002.

[4] T. Itoh and S. Tsujii, "Structure of Parallel Multipliers for a Class of Fields $GF(2^m)$", *Information and Computation*, Vol. 83, No. 1 pp. 21–40, 1989.

[5] G. Drolet, "A New Representation of Elements of Finite Fields $GF(2^m)$ Yielding Small Complexity Arithmetic Circuits", *IEEE Trans. on Computers*, Vol. 47, No. 9, pp. 938–946, Sept. 1998.

[6] W. Geiselmann, J. Müller-Quade, and R. Steinwandt, "On 'A New Representation of Elements of Finite Fields $GF(2^m)$ Yielding Small Complexity Arithmetic Circuits' ", *IEEE Trans. on Computers*, Vol. 51, No. 12, pp. 1460–1461, Dec. 2002

[7] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup, "Gauss Periods and Fast Exponentiation in Finite Fields". *Lecture Notes in Computer Science*, Vol. 911, pp. 311–322, 1995.

[8] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup, "Algorithms for Exponentiation in Finite Fields". *J. Symbolic Computation*, Vol. 29, pp. 879–899, 2000.

[9] J.H. Silverman, "Low Complexity Multiplication in Rings", *Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES'99)*, pp. 122–134, 1999

[10] H. Wu, M.A. Hasan, I.F. Blake and S. Gao, "Finite Field Multiplier Using Redundant Representation", *IEEE Trans. on Computers*, Vol. 51, No. 11, pp. 1306–1315, Nov. 2002.

[11] R. Mullin, I. Onyszchuk, S.A. Vanstone, and R. Wilson, "Optimal Normal Bases in $GF(p^n)$", *Discrete Applied Math.*, Vol. 22, pp. 149–161, 1988.

[12] A.J. Menezes, *Applications of Finite Fields*, Kluwer Academic Publishers, 1993.

[13] C.C. Wang, T.K. Truong, H.M. Shao, L.J. Deutsch, J.K. Omura, and I.S. Reed, "VLSI Architectures for Computing Multiplications and Inverses in $GF(2^m)$, *IEEE Trans. on Computers*, Vol. 34, No. 8, pp. 709–717, Aug. 1985.

[14] A. Reyhani-Masoleh and M.A. Hasan, "Efficient Multiplication Beyond Optimal Normal Bases", *IEEE Trans. on Computers*, Vol. 52, No. 4, pp. 428–439, Apr. 2003.

[15] A. Reyhani-Masoleh and M.A. Hasan, "Fast Normal Basis Multiplication Using General Purpose Processors", *IEEE Trans. on Computers*, Vol. 52, No. 11, pp. 1379–1390, Nov. 2003.

[16] Ç.K. Koç and B. Sunar, "Low-Complexity Bit-Parallel Canonical and Normal Basis Multipliers for a Class of Finite Fields", *IEEE Trans. on Computers*, Vol. 47, No. 3, pp. 353–356, March. 1998.

[17] M.A. Hasan, M.Z. Wang, and V.K. Bhargava, "A Modified Massey-Omura Parallel Multiplier for a Class of Finite Fields", *IEEE Trans. Computers*, Vol. 42, No. 10, pp. 1278–1280, Oct. 1993.

[18] A. Reyhani-Masoleh and M.A Hasan, "A New Construction of Massey-Omura Parallel Multiplier over $GF(2^m)$", *IEEE Trans. Computers*, Vol. 51, No. 5, pp. 511–520, May 2002.

[19] I.F. Blake, R.M. Roth, and G. Seroussi, "Efficient Arithmetic in $GF(2^n)$ through Palindromic Representation", Hewlett-Packard, HPL-98-134, Aug. 1998.

[20] B. Sunar and Ç.K. Koç, "An Efficient Optimal Normal Basis Type II Multiplier", *IEEE Trans. on Computers*, Vol. 50, No. 1, pp. 83–87, Jan. 2001.